

APELLIDOS, Nombre:

Nº Matrícula:

UPM ETSIInf. Departamento de Lenguajes y Sistemas Informáticos e Ingeniería del Software.
Examen de Programación II. 16 de Diciembre de 2013. Segundo Parcial.

Realización: El test se realizará en la hoja de respuesta. Es **importante** que no olvidéis rellenar vuestros datos personales y el código clave de vuestro enunciado. Se pueden utilizar hojas aparte como borrador.

Duración: La duración total del test será de **50 minutos**.

Puntuación: El test se valora sobre **10 puntos**. Las preguntas tipo test pueden tener una única respuesta o varias respuestas, el enunciado lo deja claro. Cada pregunta con una única respuesta respondida correctamente vale 1 punto, e incorrectamente respondida resta 1/3 puntos. Si en una pregunta con una única respuesta se selecciona más de una respuesta, la pregunta se puntuará con 0 puntos. Para una pregunta con varias respuestas, cada afirmación correcta seleccionada suma $1/n^{\circ}_{respuestas_correctas}$ puntos, y cada afirmación incorrecta seleccionada resta $1/n^{\circ}_{respuestas_correctas}$ puntos. Las preguntas no contestadas suman 0 puntos en cualquier caso.

Calificaciones: Las calificaciones se publicarán en Moodle como muy tarde el día **18 de Diciembre de 2013**

Revisión: Las revisiones serán el día **19 de Diciembre de 2013** de 16:00h a 18:00h en el despacho D-2315.

Primer Ejercicio

En relación a las clases y métodos abstractos en Java.

Pregunta 1

Señalar **todas** las afirmaciones verdaderas. Puede haber más de una afirmación correcta.

- A)** Una clase con algún método abstracto debe ser declarada como abstracta.
 - B)** Una clase abstracta sólo puede contener métodos abstractos.
 - C)** De una clase abstracta se puede crear una instancia, aunque con la instancia no se pueden usar los métodos abstractos.
 - D)** Las clases abstractas sirven para declarar métodos comunes a varias clases pero que están implementados de manera diferente en cada clase.
-

Segundo Ejercicio

Dadas las siguientes clases, cada una definida en su fichero .java correspondiente:

```
public abstract class ClaseAbstracta {
    public abstract void metodo();
}

public abstract class ClaseAbstractaHija extends ClaseAbstracta {
    public void metodo() {
        try {
            boolean[] v = new boolean[10];
            v[33] = false;
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.print("A_");
        }
        throw new NullPointerException();
    }

    public abstract void metodoDos();
}
```

Pregunta 2

Señalar la afirmación que es verdadera. Sólo una respuesta es correcta.

- A)** Las clases ClaseAbstracta y ClaseAbstractaHija no contienen errores sintácticos, es decir, compilan sin problemas.
 - B)** La clase ClaseAbstracta no compila porque no contiene una implementación para el método 'metodo'.
 - C)** La clase ClaseAbstractaHija no compila porque una clase que hereda de una clase abstracta no puede ser abstracta.
 - D)** La clase ClaseAbstractaHija no compila porque contiene el método metodoDos que es abstracto.
-

Tercer Ejercicio

Dada la siguiente definición del metodo factorial en una clase llamada Factorial:

```
public int factorial(int num) throws ExcepcionNNegativo {
    if (num < 0) {
        throw new ExcepcionNNegativo(" el metodo no admite numeros negativos");
    }
    if (num == 0 || num == 1) {
        return 1;
    } else {
        return num * factorial(num - 1);
    }
}
```

Pregunta 3

Indicar cuál es la salida por consola del siguiente código.

```
public static void main(String args[]) throws ExcepcionNNegativo {
    Factorial ejemplo = new Factorial();
    try {
        System.out.print(ejemplo.factorial(-3));
        System.out.print(" Se produjo un error ");
        System.out.print(ejemplo.factorial(3));
    } catch (ExcepcionNNegativo e) {
        System.out.print(" Error ");
        System.out.print(ejemplo.factorial(0));
    }
}
```

Sólo una respuesta es correcta.

- A) Error 0
- B) Se produjo un error 6Error 1
- C) Error 1**
- D) Se produjo un error 6

Cuarto Ejercicio

Sean las siguientes afirmaciones sobre aserciones y excepciones en java.

Pregunta 4

Señalar **todas** las afirmaciones verdaderas. Puede haber más de una afirmación correcta.

- A) La herramienta JUnit no usa ninguna forma de aserciones.
- B) Las excepciones no paran la ejecución normal de un programa.
- C) Las aserciones sirven para depurar el código.**
- D) Las excepciones sirven para notificar una situación anómala.**

Quinto Ejercicio

Sean las siguientes afirmaciones sobre las clases contenedoras:

Pregunta 5

Señalar la afirmación que es verdadera. Sólo una respuesta es correcta.

- A) En la clase contenedora Pila sólo se puede acceder al elemento más antiguo en la pila.
- B) En la clase contenedora Cola existen operaciones para acceder directamente a cualquier elemento de la cola.
- C) Las clases contenedoras se usan para almacenar datos de manera estructurada.**
- D) Las clases contenedoras se usan para almacenar datos de manera no estructurada.

Sexto Ejercicio

Dada la siguiente clase donde se supone que se han importado las clases necesarias:

```
public class EjercicioCola {
    public static void main(String args[]) throws ExcepcionColaVacia {
        ColaSimple<String> cola = new ColaSimple<String>();
        Scanner lector = new Scanner(System.in);
        String entrada;

        // Leer strings separados por espacio en blanco
        while (lector.hasNext()) {
            entrada = lector.next(); // lee el siguiente string
            if (entrada.length() > 1) {
                cola.insertar(entrada);
            } else if (entrada.length() == 1) {
                if (cola.estaVacia()) {
                    System.out.print("No hay nadie esperando");
                } else {
                    System.out.print(cola.sacarPrimero() + ", acuda a la mesa "
                        + entrada + " ");
                }
            }
        }
        while (!cola.estaVacia()) {
            System.out.print(cola.sacarPrimero() + ", vuelva mañana");
        }
    }
}
```

Pregunta 6

Indicar cuál será el resultado de ejecutar el main() con la siguiente entrada por consola "Clara Luis A A B".

Sólo una respuesta es correcta.

- A)** Clara, acuda a mesa A Luis, acuda a mesa A No hay nadie esperando
- B)** Luis, acuda a mesa A Clara, acuda a mesa A
- C)** Clara, acuda a mesa A Luis, vuelva mañana
- D)** Luis, acuda a mesa A Clara, acuda a mesa A No hay nadie esperando

Séptimo Ejercicio

Indicar cuáles de las siguientes afirmaciones sobre las clases contenedoras y su implementación son ciertas:

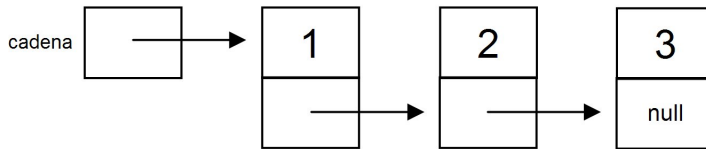
Pregunta 7

Señalar **todas** las afirmaciones verdaderas. Puede haber más de una afirmación correcta.

- A)** Las pilas acotadas deben contener en todo momento el número máximo de elementos.
- B)** Las listas se usan para almacenar datos sin que se destruya la lista al recorrerla.
- C)** Los elementos que se guardan en una cola permanecen en la cola al procesar la misma.
- D)** El contenedor pila se suele implementar con una cadena enlazada excepto si es una pila acotada en cuyo caso es mejor implementarla con un **array**.

Octavo Ejercicio

Dada la siguiente cadena simplemente enlazada de nodos:



Creada a partir de la siguiente clase genérica *Nodo*:

```

public class Nodo<Informacion> {
    private Nodo<Informacion> siguiente = null;
    private Informacion dato;
    public Nodo(Informacion dato) {
        this.dato = dato;
    }
    public Nodo(Informacion dato, Nodo<Informacion> siguiente) {
        this.dato = dato;
        this.siguiente = siguiente;
    }
    public Informacion getDato() {
        return this.dato;
    }
    public Nodo<Informacion> getSiguiente() {
        return this.siguiente;
    }
    public void setSiguiente(Nodo<Informacion> siguiente) {
        this.siguiente = siguiente;
    }
}
  
```

Pregunta 8

Indicar cuál de las siguientes secuencias de sentencias permite insertar en la cadena un objeto

`Nodo nodo = new Nodo<Double>(1.5)`

entre el nodo 1 y el nodo 2, dejando el resto de los demás nodos en el mismo orden y accesibles a través de la referencia *cadena*. **Sólo una respuesta es correcta.**

- A) `nodo.setSiguiente(cadena.getSiguiente());`
- B) `cadena.setSiguiente(nodo); nodo.setSiguiente(cadena.getSiguiente());`
- C) `nodo.setSiguiente(cadena.getSiguiente().getSiguiente()); cadena.setSiguiente(nodo);`
- D) `nodo.setSiguiente(cadena.getSiguiente()); cadena.setSiguiente(nodo);`**

Noveno Ejercicio

Indicar cuáles de las siguientes afirmaciones sobre las clases genéricas y envoltorio son ciertas:

Pregunta 9

Señalar **todas** las afirmaciones verdaderas. Puede haber más de una afirmación correcta.

- A) Las clases contenedoras nunca son clases genéricas.
- B) Las clases envoltorio se usan para poder tratar datos primitivos como objetos**
- C) Las clases genéricas se usan para poder tratar objetos como datos primitivos.
- D) Las clases genéricas permiten parametrizar el tipo de contenido de una clase contenedora.**

Décimo Ejercicio

En relación a las pruebas unitarias (*unit tests*).

Pregunta 10

Señalar la afirmación que es verdadera. Sólo una respuesta es correcta.

- A) Sólo se ejecutan una vez, y el código de prueba no se conserva
- B) Su objetivo es probar todo el sistema en su conjunto
- C) Son pruebas que tienen que ejecutarse manualmente
- D) Su objetivo es probar un elemento del sistema (paquete, clase ,método, ...)**